

Görev-5

ASP.NET Core WebAPI Blog Uygulaması

Amaç

Bu ödevde, basit bir ASP.NET Core Web API projesi geliştirmeniz istenmektedir. Projede bir blog uygulaması yönetilecek ve bu blog uygulaması üzerinde temel CRUD (Create, Read, Update, Delete) işlemlerini gerçekleştirecek bir API oluşturacaksınız. Endpoint'lerin beklediği request ve döndürdüğü response tiplerini belirleyerek API endpoint'lerine produce ve consume özellikleri ekleyeceksiniz. Ayrıca endpoint'leriniz için bir Postman collection'ı oluşturacaksınız.

Not: Uygulama içerisindeki makaleler static bir listede tutulacaktır.

Makale Class'ı :

```
public class Article
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }
}
```

Not: Her bir makalenin **Id** özelliği bulunmaktadır. Bu özellik ilgili elemanın liste içerisindeki index'inin 1 fazlası olacaktır. **Id** sadece makale kaydedilirken atanıp, sonradan değiştirilmeyecektir.

API Ayrıntıları

1. Makaleleri Listeleme (GET)

- Endpoint: /api/articles
- İstenen Veri Türü: JSON
- İşlev: Tüm makaleleri listeler.
- ProducesResponseType: 200 OK (Makale listesi)
- Consumes: Yok

Örnek Response Body:

```
[
  {
    "id": 1,
    "title": "ASP.NET Core ile Web API Geliştirme",
    "content": "Bu makalede ASP.NET Core ile Web API geliştirmenin temellerini öğreneceksiniz."
  },
  {
    "id": 2,
    "title": "Postman Kullanımı",
    "content": "Postman ile API testlerini nasıl yapabileceğinizi öğreneceksiniz."
  }
]
```

○

2. Makale Detayını Getirme (GET)

- Endpoint: /api/articles/{id}
- İstenen Veri Türü: JSON
- İşlev: Belirli bir makalenin detaylarını getirir.
- ProducesResponseType: 200 OK (Makale detayı), 404 Not Found
- Consumes: Yok

Örnek Response Body:

```
{
  "id": 1,
  "title": "ASP.NET Core ile Web API Geliştirme",
  "content": "Bu makalede ASP.NET Core ile Web API geliştirmenin temellerini öğreneceksiniz."
}
```

```
}
```

3. Makale Ekleme (POST)

- Endpoint: /api/articles
- İstenen Veri Türü: JSON (Makale nesnesi)
- İşlev: Yeni bir makale oluşturur, id'sini belirler ve ekler.
- ProducesResponseType: 201 Created (Yeni makale oluşturuldu), 400 Bad Request (Makale title'ı boş veya null ise)
- Consumes: JSON

Örnek Request Body:

```
{  
  "title": "Yeni Makale",  
  "content": "Bu bir örnek makaledir."  
}
```

4. Makale Güncelleme (PUT)

- Endpoint: /api/articles/{id}
- İstenen Veri Türü: JSON (Makale nesnesi)
- İşlev: Belirli bir makalenin içeriğini günceller.
- ProducesResponseType: 200 OK (Makale güncellendi), 400 Bad Request (Makale title'ı boş veya null ise)
- Consumes: JSON

Örnek Request Body:

```
{  
  "title": "Güncellenmiş Makale",  
  "content": "Bu makale güncellendi."  
}
```

5. Makale Silme (DELETE)

- Endpoint: /api/articles/{id}

- İşlev: Belirli bir makaleyi siler.
- ProducesResponseType: 200 OK (Makale silindi), 204 No Content (Makale zaten mevcut değildi)
- Consumes: Yok

6. Postman Collection Oluşturma

- API'yi test etmek için Postman koleksiyonu oluşturun.
- Koleksiyon içinde yukarıdaki tüm istekleri ekleyin.

Teslim: WebAPI Uygulamasını içeren klasörü zip'leyin. Ayrı bir klasör içerisinde de Postman collection'ınızın export edilmiş json dosyalarını zip'leyerek classroom'da Görev-5 paylaşımına ek olarak ekleyin. WebAPI ve Collection'ları içeren zip dosyaları ayrı ayrı olmalıdır, toplamda iki adet zip dosyası teslim etmeniz gerekmektedir.